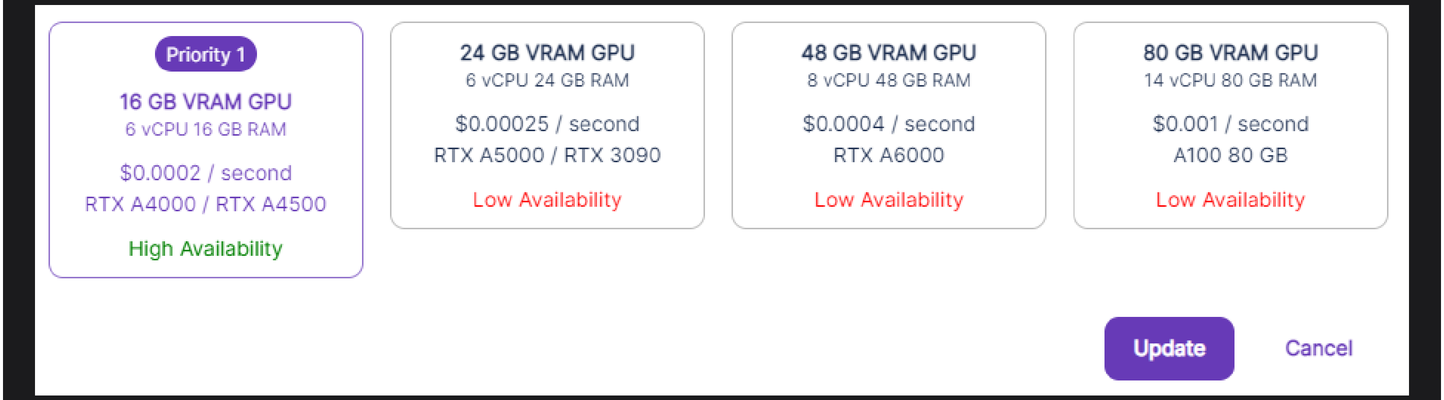
Test response time

When setting up an API, you have several options available at different price points and resource allocations. You can select a single option if you would prefer to only use one price point, or select a preference order between the pools that will allocate your requests accordingly.



Priority 1
16 GB VRAM GPU
6 vCPU 16 GB RAM
$0.0002 / second
RTX A4000 / RTX A4500
High Availability

24 GB VRAM GPU
6 vCPU 24 GB RAM
$0.00025 / second
RTX A5000 / RTX 3090
Low Availability

48 GB VRAM GPU
8 vCPU 48 GB RAM
$0.0004 / second
RTX A6000
Low Availability

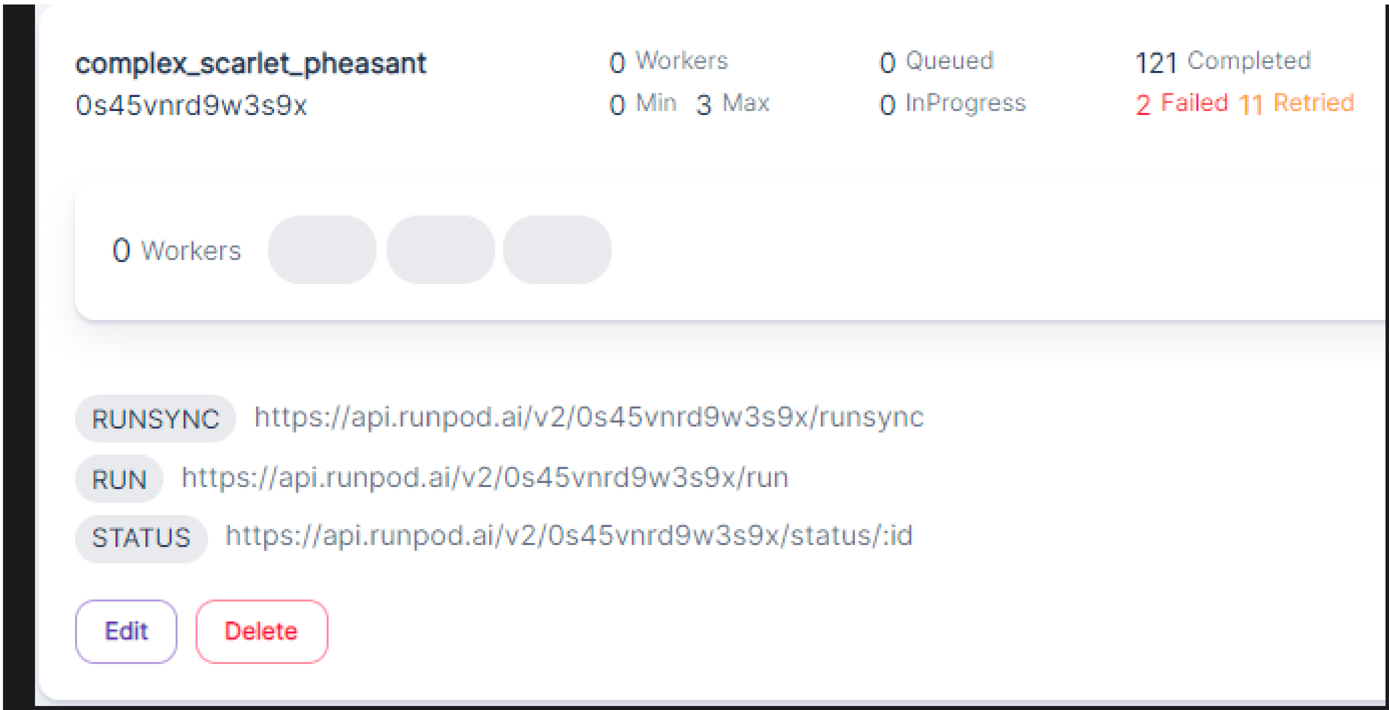
80 GB VRAM GPU
14 vCPU 80 GB RAM
$0.001 / second
A100 80 GB
Low Availability

The option that will be most cost effective for you will be based on your use case and your tolerance for task run time. Each situation will be different, so when deciding which API to use, it's worth it to do some testing to not only find out how long your tasks will take to run, but how much you might expect to pay for each task.

To find out how long a task will take to run, select a single pool type as shown in the image above. Then, you can send a request to the API through your preferred method. If you're unfamiliar with how to do so or don't have your own method, then you can use a free option like reqbin.com to send an API request to the RunPod severs.

The URLs to use in the API will be shown in the My APIs screen:

Ask AI



complex_scarlet_pheasant
0s45vnrd9w3s9x

0 Workers
0 Min 3 Max

0 Queued
0 InProgress

121 Completed
2 Failed 11 Retried

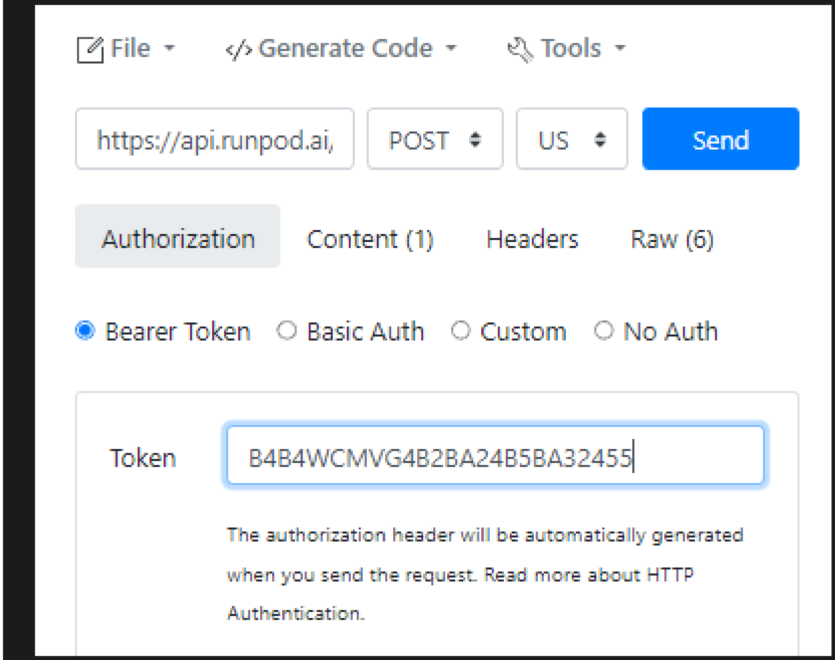
0 Workers

RUNSYNC https://api.runpod.ai/v2/0s45vnrd9w3s9x/runsync

RUN https://api.runpod.ai/v2/0s45vnrd9w3s9x/run

STATUS https://api.runpod.ai/v2/0s45vnrd9w3s9x/status/:id

On reqbin.com, enter the Run URL of your API, select POST under the dropdown, and enter your API key that was given when you created the key under Settings(if you do not have it saved, you will need to return to Settings and create a new key). Under Content, you will also need to give it a basic command (in this example, we've used a Stable Diffusion prompt).

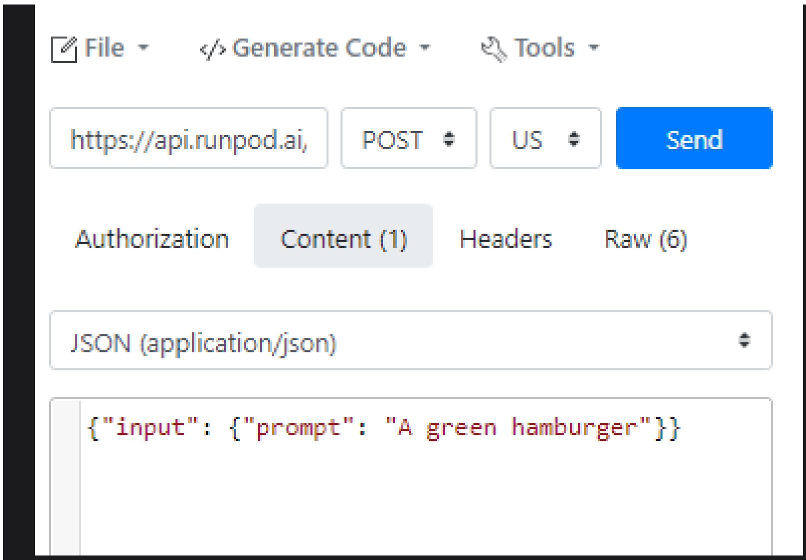


https://api.runpod.ai, POST US

● Bearer Token ○ Basic Auth ○ Custom ○ No Auth

Token B4B4WCMVG4B2BA24B5BA32455

The authorization header will be automatically generated when you send the request. Read more about HTTP Authentication.



https://api.runpod.ai,

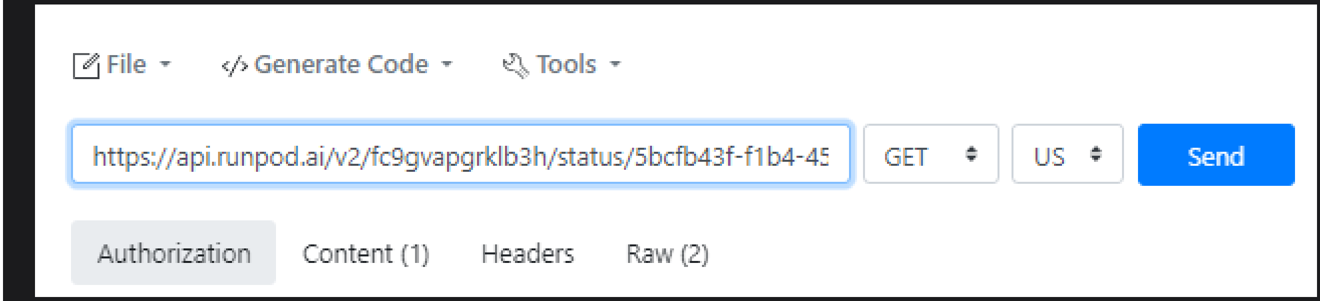
POST

US

JSON (application/json)

```
{"input": {"prompt": "A green hamburger"}}
```

Send the request, and it will give you an ID for the request and notify you that it is processing. You can then swap the URL in the request field with the Status address and add the ID to the end of it, and click Send.

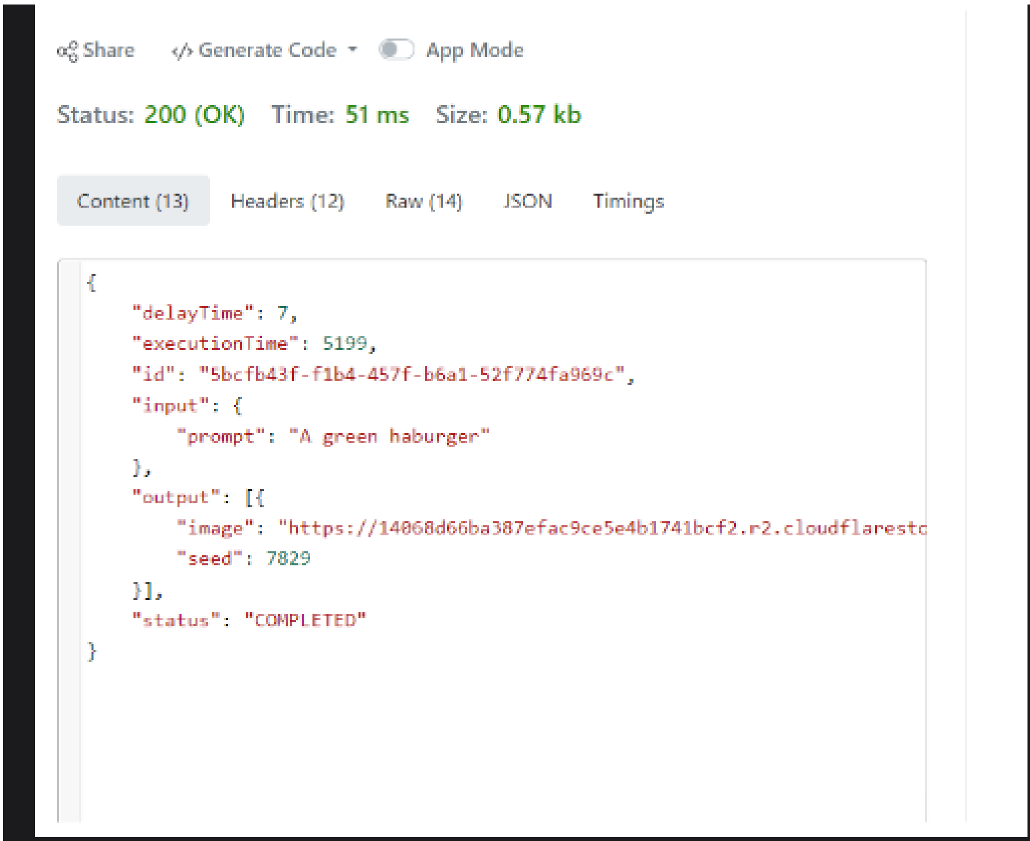


https://api.runpod.ai/v2/fc9gvapgrklb3h/status/5bcfb43f-f1b4-45

GET

US

It will return a Delay Time and an Execution Time, denoted in milliseconds. The Delay Time should be extremely minimal, unless the API process was spun up from a cold start, then a sizable delay is expected for the first request sent. The Execution Time is how long the GPU took to actually process the request once it was received. It may be a good idea to send a number of tests so you can get a min, max, and average run time -- five tests should be an adequate sample size.



Status: 200 (OK) Time: 51 ms Size: 0.57 kb

```
{
    "delayTime": 7,
    "executionTime": 5199,
    "id": "5bcfb43f-f1b4-457f-b6a1-52f774fa969c",
    "input": {
        "prompt": "A green haburger"
    },
    "output": [{
        "image": "https://14068d66ba387efac9ce5e4b1741bcf2.r2.cloudflarestc
        "seed": 7829
    }],
    "status": "COMPLETED"
}
```

You can then switch the GPU pool above to a different pool and repeat the process.

What will ultimately be right for your use case will be determined by how long you can afford to let the process run. For heavier jobs, a task on a slower GPU will be likely be more cost-effective with a tradeoff of speed. For simpler tasks, there may also be diminishing returns on how fast the task that can be run that may not be significantly improved by selecting higher-end GPUs. Experiment to find the best balance for your scenario.

Edit this page

Previous
« Use environment variables

Next
Deploy »